```php
<?php

/**
 * @author GOLAY Brian
 * @version 1.0 (2021/05/20)
 * Group-related functions
 */

namespace FlightClub\sql;

use FlightClub\sql\DBConnection;

require_once 'dbConnection.php';

/**
 * Group related class that contain group related functions with database interraction
 */
class GroupDAO
{
    /**
     * return all pending invites
     *
     * @return array[mixed] pending invites data
     */
    public static function getAllPendingInvites()
    {
        $db = DBConnection::getConnection();
        $sql = "SELECT * FROM `tbl_user-group` WHERE `Dttm_Membership` is null";
        $request = $db->prepare($sql);
        $request->execute();
        return $request->fetchAll();
    }

    /**
     * return all group data with the group Id
     *
     * @param int $idGroup
     * @return array[mixed] all group data
     */
    public static function getGroupById($idGroup)
    {
        $db = DBConnection::getConnection();
        $sql = "SELECT * FROM `tbl_group` WHERE `Id_Group` = :id";
        $request = $db->prepare($sql);
        $request->execute([
            ':id' => $idGroup
        ]);
        return $request->fetch();
    }

    /**
     * Delete a pending invite
     *
     * @param int $idGroup
     * @param int $idUser
     * @return void
     */
    public static function deletePendingInvite($idGroup, $idUser)
    {
        $db = DBConnection::getConnection();
        $sql = "DELETE FROM `tbl_user-group` WHERE `Id_Group` = :idGroup AND `Id_User` = :idUser";

        $request = $db->prepare($sql);
        $request->execute([
            ":idGroup" => $idGroup,
            ":idUser" => $idUser
        ]);
    }

    /**
     * Accept a pending invite
     *
     * @param int $idGroup
     * @param int $idUser
     * @return void
     */
    public static function acceptPendingInvite($idGroup, $idUser)
    {
        $dateTime = date("Y-m-d H:i:s");
        $db = DBConnection::getConnection();
        $sql = "UPDATE `tbl_user-group` SET `Dttm_Membership` = :date WHERE `tbl_user-group`.`Id_User` = :idUser AND `tbl_user-group`.`Id_Group` = :idGroup;";

        $request = $db->prepare($sql);
        $request->execute([
            ":idGroup" => $idGroup,
            ":idUser" => $idUser,
            ":date" => $dateTime
        ]);
    }

    /**
     * Get all groups of one user identified by his id
     *
     * @param int $id
     * @return array[mixed]
     */
    public static function getAllOfMyGroup($id)
    {
        $db = DBConnection::getConnection();
        $sql = "SELECT * FROM `tbl_user-group` join `tbl_group` on `tbl_user-group`.`Id_Group` = `tbl_group`.`Id_Group`  WHERE `Dttm_Membership` is not null AND `tbl_user-group`.`Id_User` = :id";
        $request = $db->prepare($sql);
        $request->execute([
            ':id' => $id
        ]);
        return $request->fetchAll();
    }

    /**
     * Get all users of one group identified by the group id
     *
     * @param int $idGroup
     * @return array[mixed]
     */
    public static function getAllOfGroupUser($idGroup)
    {
        $db = DBConnection::getConnection();
        $sql = "SELECT * FROM `tbl_user-group` join `tbl_user` on `tbl_user-group`.`Id_User` = `tbl_user`.`Id_User`  WHERE `Dttm_Membership` is not null AND `tbl_user-group`.`Id_Group` = :id";
        $request = $db->prepare($sql);
        $request->execute([
            ':id' => $idGroup
        ]);
        return $request->fetchAll();
    }

    /**
     * Get an users of one group identified by his id
     *
     * @param int $idGroup
     * @param int $idUser
     * @return array[mixed]
     */
    public static function getGroupUserByIdUserAndIdGroup($idUser, $idGroup)
    {
        $db = DBConnection::getConnection();
        $sql = "SELECT * FROM `tbl_user-group` join `tbl_user` on `tbl_user-group`.`Id_User` = `tbl_user`.`Id_User`  WHERE `Dttm_Membership` is not null AND `tbl_user-group`.`Id_Group` = :idGroup AND `tbl_user-group`.`Id_User` = :idUser";
        $request = $db->prepare($sql);
        $request->execute([
            ':iduser' => $idUser,
            ':idGroup' => $idGroup
        ]);
        return $request->fetchAll();
    }

    /**
     * Create a group
     *
     * @param string $name
     * @return void
     */
    public static function createGroup($name)
    {
        $db = DBConnection::getConnection();
        $sql = "INSERT INTO `tbl_group` (`Id_Group`, `Nm_Group`) VALUES (NULL, :name);";

        $request = $db->prepare($sql);
        $request->execute([
            ":name" => $name
        ]);
    }

    /**
     * return all group data sort by the highest group id descending
     *
     * @return array[mixed]
     */
    public static function getLastGroupId()
    {
        $db = DBConnection::getConnection();
        $sql = "SELECT * FROM `tbl_group` order by `Id_Group` desc";
        $request = $db->prepare($sql);
        $request->execute();
        return $request->fetchAll();
    }

    /**
     * Join a group just created by the user
     *
     * @param int $idGroup
     * @param int $idUser
     * @return void
     */
    public static function joinCreatedGroup($idGroup, $idUser)
    {
        $dateTime = date("Y-m-d H:i:s");
        $db = DBConnection::getConnection();
        $sql = "INSERT INTO `tbl_user-group` (`Id_User`, `Id_Group`, `Dttm_Invitation`, `Dttm_Membership`) VALUES (:idUser, :idGroup, :date, :date);";

        $request = $db->prepare($sql);
        $request->execute([
            ":idGroup" => $idGroup,
            ":idUser" => $idUser,
            ":date" => $dateTime
        ]);
    }


    /**
     * Create and send an invitation to an user
     *
     * @param int $idGroup
     * @param int $idUser
     * @return void
     */
    public static function CreateInvite($idGroup, $idUser)
    {
        $dateTime = date("Y-m-d H:i:s");
        $db = DBConnection::getConnection();
        $sql = "INSERT INTO `tbl_user-group` (`Id_User`, `Id_Group`, `Dttm_Invitation`, `Dttm_Membership`) VALUES (:idUser, :idGroup, :date, null);";

        $request = $db->prepare($sql);
        $request->execute([
            ":idGroup" => $idGroup,
            ":idUser" => $idUser,
            ":date" => $dateTime
        ]);
    }


    /**
     * return the user group data with the group Id and user id
     *
     * @param int $idGroup
     * @return array[mixed] all group data
     */
    public static function getMyGroupById($idGroup, $idUser)
    {
        $db = DBConnection::getConnection();
        $sql = "SELECT * FROM `tbl_user-group` WHERE `Id_Group` = :idGroup AND `Id_User` = :idUser AND Dttm_Membership is not null";
        $request = $db->prepare($sql);
        $request->execute([
            ':idGroup' => $idGroup,
            ':idUser' => $idUser
        ]);
        return $request->fetch();
    }
}

```